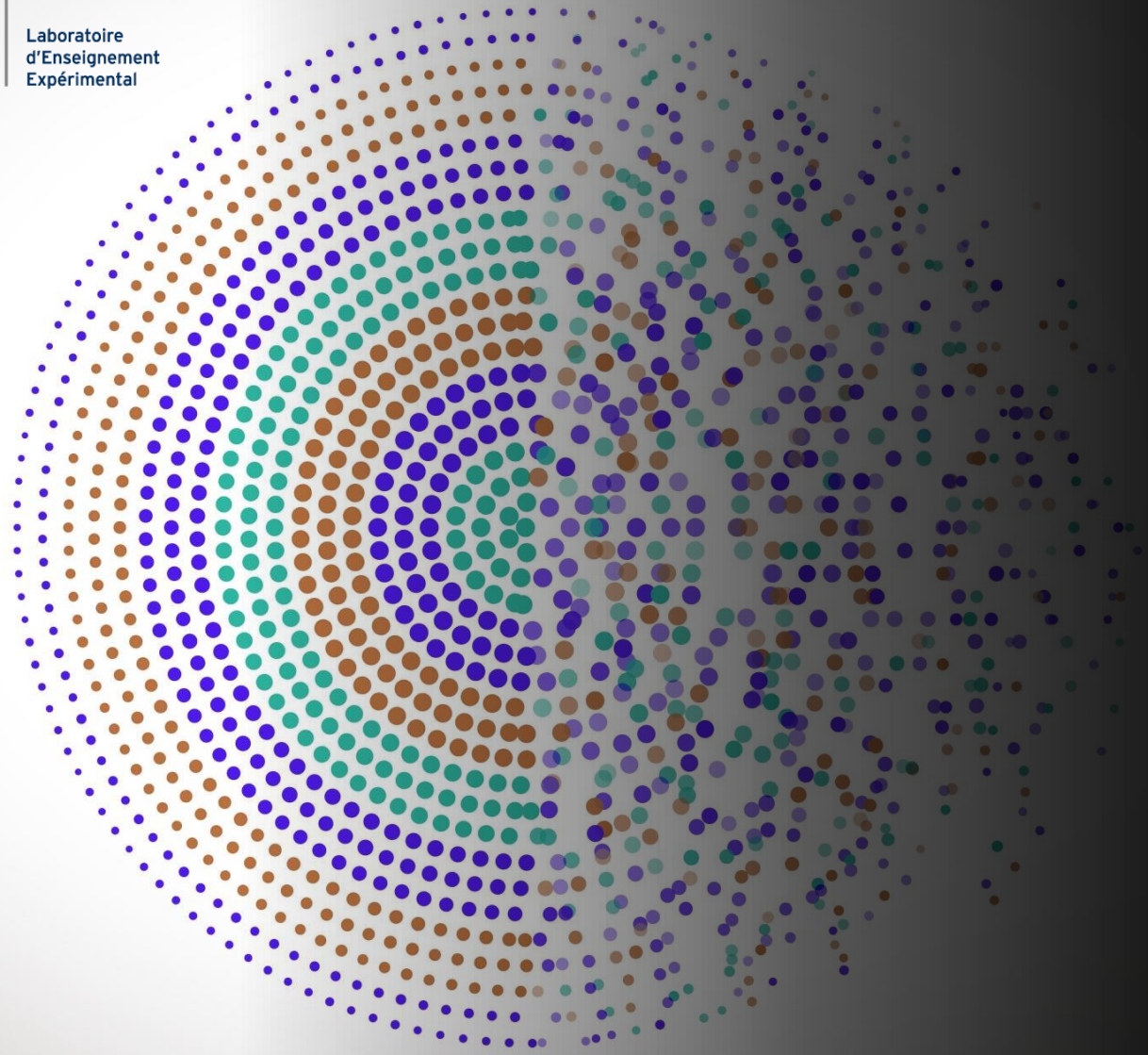
INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental



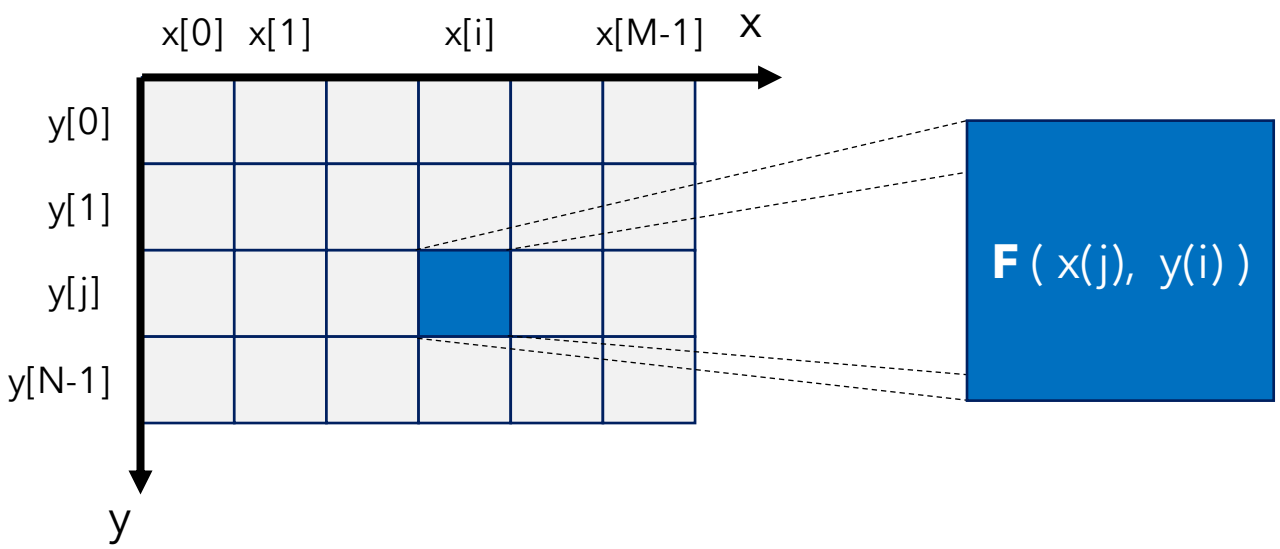
Python / Numpy

Meshgrid

Digital Methods
Institut d'Optique / Notions

How to fill a 2D array ?

- Problem :
  - Fill a **2D array** with a specific value depending on x-axis and y-axis index



- For example :

```
def F(a, b):
    return a + b
```

- We assume that x and y are defined :

```
x = np.arange(…) # length = M
y = np.arange(…) # length = N
```

How to fill a 2D array ?

- First idea : a **double loop** on i and j

```
output_array = np.zeros((N, M))

for i in range(N):
     for j in range(M):
          output_array[i][j] = F(x[j], y[i])
```

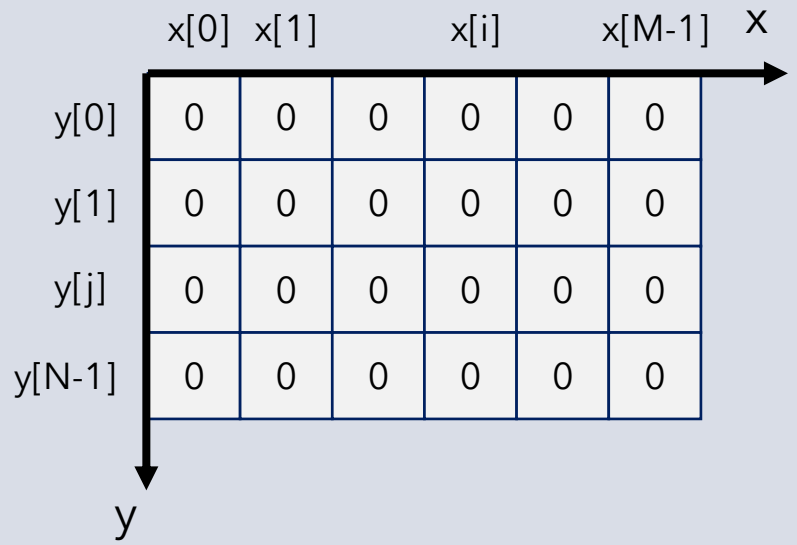
How to fill a 2D array ?

- Example (N and M are integers) :

```
output_array = np.zeros((N, M))

for i in range(N):
     for j in range(M):
          output_array[i][j] = F(x[j], y[i])
```

```
x = np.linspace(0, M-1, M)
y = np.linspace(0, N-1, N)
```

```
def F(a, b):
    return a + b
```



How to fill a 2D array ?

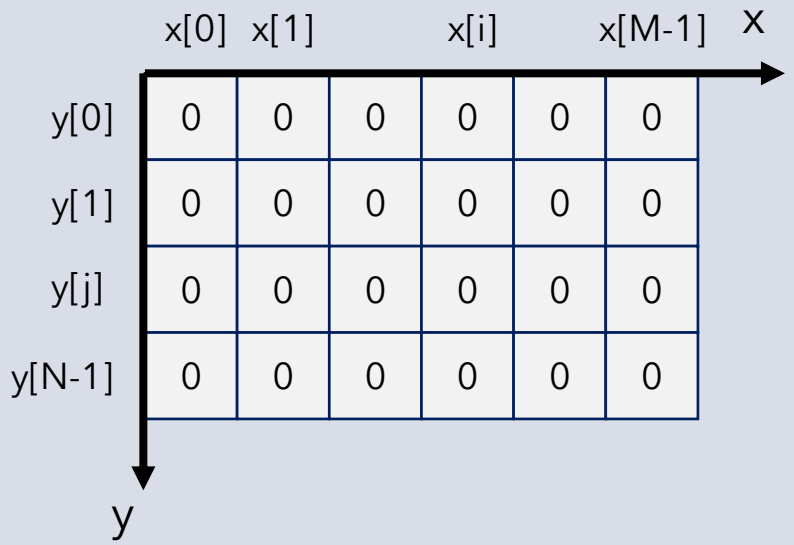
- Example (N and M are integers) :

```
output_array = np.zeros((N, M))

for i in range(N):
     for j in range(M):
          output_array[i][j] = F(x[j], y[i])
```

i = **0**

```
x = np.linspace(0, M-1, M)
y = np.linspace(0, N-1, N)
```

```
def F(a, b):
    return a + b
```



How to fill a 2D array ?

- Example (N and M are integers) :

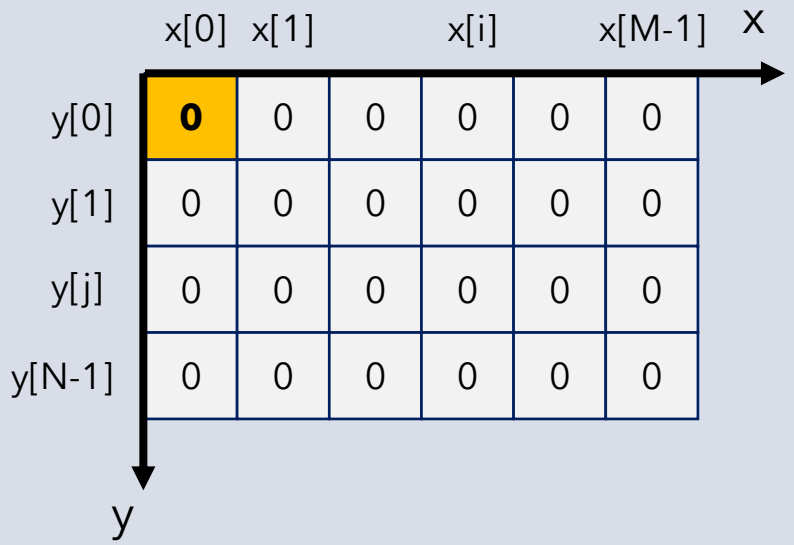
```
output_array = np.zeros((N, M))

for i in range(N):
    for j in range(M):
        output_array[i][j] = F(x[j], y[i])
```

```
x = np.linspace(0, M-1, M)
y = np.linspace(0, N-1, N)
```

```
def F(a, b):
    return a + b
```

i = **0**

j = 0 → ouput_array[**0**][0] = F(x[0], y[**0**]) = 0



How to fill a 2D array ?

- Example (N and M are integers) :

```
output_array = np.zeros((N, M))

for i in range(N):
    for j in range(M):
        output_array[i][j] = F(x[j], y[i])
```

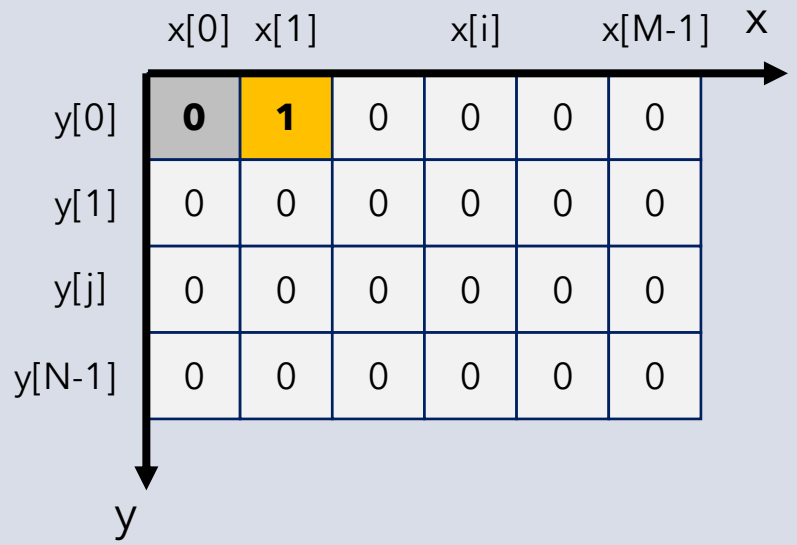
i = **0**

j = 0 → ouput_array[**0**][0] = F(x[0], y[**0**]) = 0

j = 1 → ouput_array[**0**][1] = F(x[1], y[**0**]) = 1

```
x = np.linspace(0, M-1, M)
y = np.linspace(0, N-1, N)
```

```
def F(a, b):
    return a + b
```



How to fill a 2D array ?

- Example (N and M are integers) :

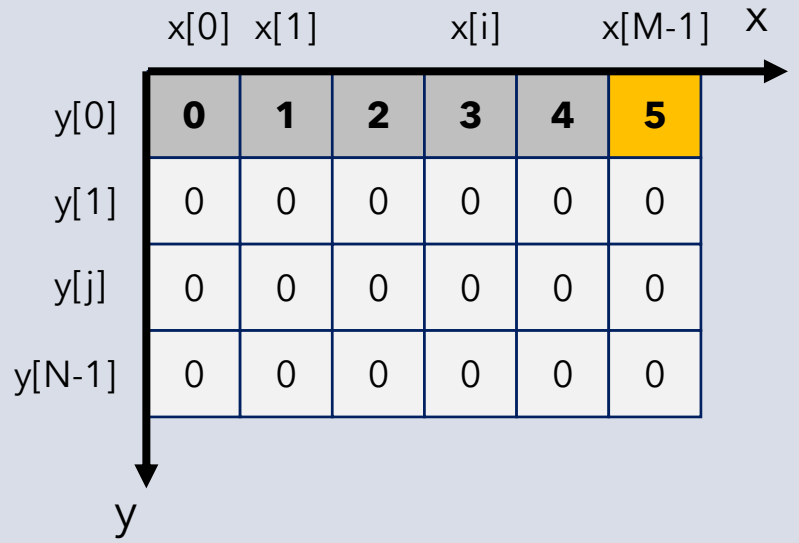
```
output_array = np.zeros((N, M))

for i in range(N):
    for j in range(M):
        output_array[i][j] = F(x[j], y[i])
```

```
x = np.linspace(0, M-1, M)
y = np.linspace(0, N-1, N)
```

```
def F(a, b):
    return a + b
```

i = **0**

j = 0 → ouput_array[**0**][0] = F(x[0], y[**0**]) = 0
j = 1 → ouput_array[**0**][1] = F(x[1], y[**0**]) = 1
j = 2 → ouput_array[**0**][2] = F(x[2], y[**0**]) = 2
j = 3 → ouput_array[**0**][3] = F(x[3], y[**0**]) = 3
j = 4 → ouput_array[**0**][4] = F(x[4], y[**0**]) = 4
j = 5 → ouput_array[**0**][5] = F(x[5], y[**0**]) = 5



How to fill a 2D array ?

- Example (N and M are integers) :

```
output_array = np.zeros((N, M))

for i in range(N):
    for j in range(M):
        output_array[i][j] = F(x[j], y[i])
```

```
x = np.linspace(0, M-1, M)
y = np.linspace(0, N-1, N)
```

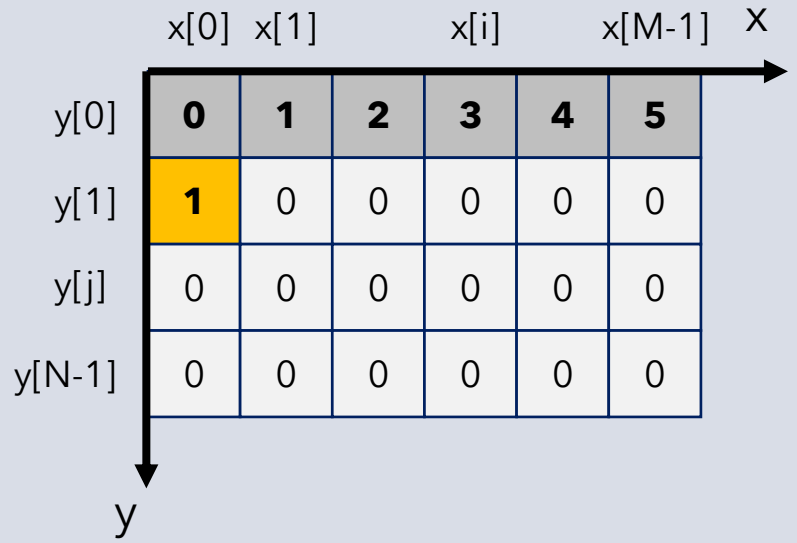
```
def F(a, b):
    return a + b
```

i = **0**

…

i = **1**

j = 0 → ouput_array[**1**][0] = F(x[0], y[**1**]) = 1



How to fill a 2D array ?

- Example (N and M are integers) :

```
output_array = np.zeros((N, M))

for i in range(N):
    for j in range(M):
        output_array[i][j] = F(x[j], y[i])
```

```
x = np.linspace(0, M-1, M)
y = np.linspace(0, N-1, N)
```

```
def F(a, b):
    return a + b
```

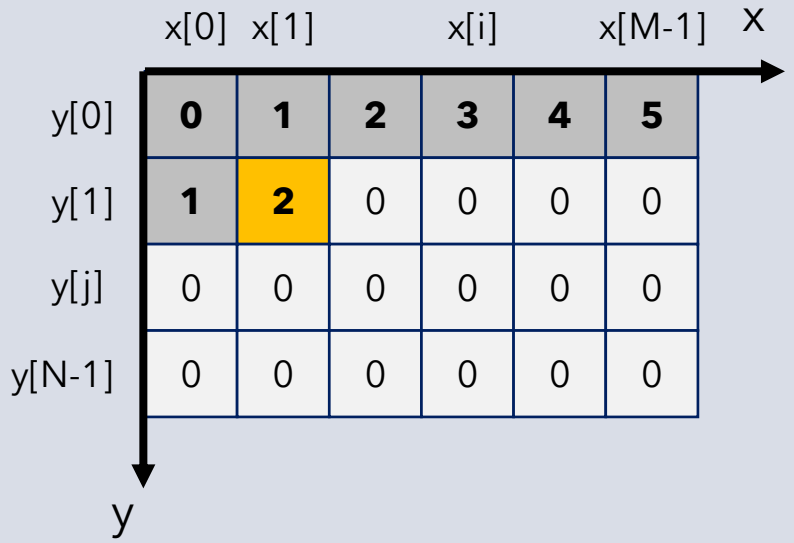
i = **0**

…

i = **1**

j = 0 → ouput_array[**1**][0] = F(x[0], y[**1**]) = 1

j = 1 → ouput_array[**1**][1] = F(x[1], y[**1**]) = 2



How to fill a 2D array ?

INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

- Second method : using **Numpy arrays** methods (***meshgrid***)

```
XX, YY = np.meshgrid(x, y)

output_array = F(YY, XX)
```

How to fill a 2D array ?

- Example (N and M are integers) :

```
XX, YY = np.meshgrid(x, y)

output_array = F(YY, XX)
```

```
x = np.linspace(0, M-1, M)
y = np.linspace(0, N-1, N)
```

```
def F(a, b):
    return a + b
```

XX

x[0]	x[1]		x[j]		x[M-1]
x[0]	x[1]		x[j]		x[M-1]
x[0]	x[1]		x[j]		x[M-1]
x[0]	x[1]		x[j]		x[M-1]

YY

y[0]	y[0]		y[0]		y[0]
y[1]	y[1]		y[1]		y[1]
y[i]	y[i]		y[i]		y[i]
y[N-1]	y[N-1]		y[N-1]		y[N-1]

How to fill a 2D array ?

- Example (N and M are integers) :

```
XX, YY = np.meshgrid(x, y)

output_array = F(YY, XX)
```

```
x = np.linspace(0, M-1, M)
y = np.linspace(0, N-1, N)
```

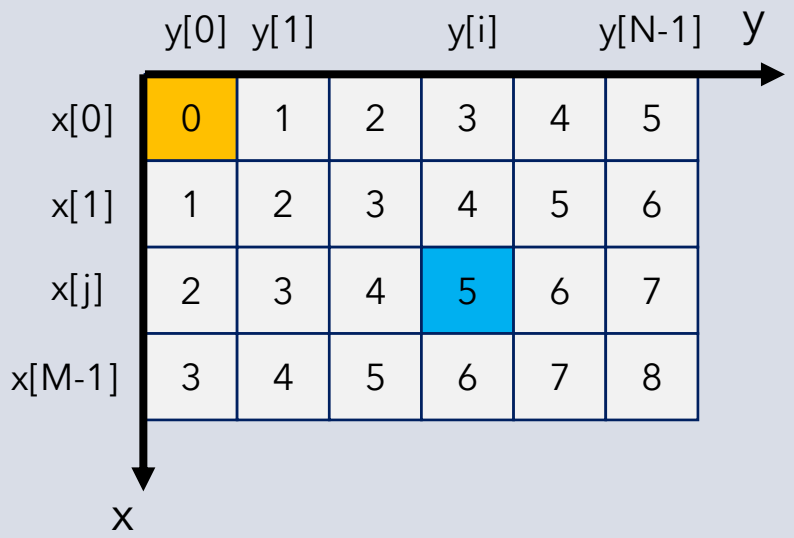
```
def F(a, b):
    return a + b
```

XX

x[0]	x[1]		x[j]		x[M-1]
x[0]	x[1]		x[j]		x[M-1]
x[0]	x[1]		x[j]		x[M-1]
x[0]	x[1]		x[j]		x[M-1]

YY

y[0]	y[0]		y[0]		y[0]
y[1]	y[1]		y[1]		y[1]
y[i]	y[i]		y[i]		y[i]
y[N-1]	y[N-1]		y[N-1]		y[N-1]



How to fill a 2D array ?

- Comparison / Execution time*

```
output_array = np.zeros((N, M))

for i in range(N):
     for j in range(M):
          output_array[i][j] = F(x[j], y[i])
```

	M=3 / N=5	M=30 / N=50	M=300 / N=500	M=3k / N=5k
	~ 9 us	690 us	70 ms	7 s
Memory Use	1 x M x N			

```
XX, YY = np.meshgrid(x, y)

output_array = F(YY, XX)
```

	~ 19 us	~ 21 us	~ 1 ms	9.4 ms
	3 x M x N			

* Executed on the same computer / Core i7 / 16 Go RAM